

NPS55GV72051A

# NAVAL POSTGRADUATE SCHOOL

// Monterey, California



PROCESSOR UTILIZATION IN MULTIPROGRAMMING SYSTEMS  
VIA DIFFUSION APPROXIMATIONS

by

Donald P. Gaver

and

Gerald S. Shedler

May 1972

Approved for public release;  
distribution unlimited.



NAVAL POSTGRADUATE SCHOOL  
Monterey, California

Rear Admiral A. S. Goodfellow, USN  
Superintendent

M. U. Clauser  
Provost

Abstract:

This report describes an approximation for the central processor (CPU) utilization in a multiprogramming computer system. The approximation is based on use of the mathematical theory of diffusion; its adequacy is checked numerically and found to be good.



## 1. INTRODUCTION.

Several authors have recognized that interesting characteristics of multiprogramming computer systems may be obtained by analysis of the cyclic queues occurring at CPU (Central Processor Unit) and DTU (Data Transmission Unit); see Gaver [3], Lewis and Shedler [5], Shedler [8], [9], and others. A system feature that complicates the probability analysis is the non-Markovian nature of DTU service: while it may be roughly plausible to assume CPU service times (e.g., times to page fault) to be independently and exponentially distributed, it is apparently much less reasonable to make such a distributional assumption about DTU service times. Although straightforward attacks on this problem have been successfully conducted, see [5] and [8], it may be useful to attempt a simpler, approximate, approach, and this is the objective of the present paper. The results obtained here have the virtue of a refreshing mathematical simplicity, but are of course only in approximate agreement with results such as those obtained in [8]. When we reflect that our present models for multiprogramming are quite simplistic in the light of current knowledge of a) program behavior at the CPU, b) information accessing at the DTU, and c) representation of storage hierarchies, approximations of this type may be quite adequate for providing insights. We envision using results of this kind to design computer systems with acceptable cost-performance characteristics. Such results are useful, for example, in suggesting performance characteristics of devices which will result in a balanced system. In addition, by taking cost characteristics

into account, one can begin to evaluate technologies from a total system point of view, and thus be guided to select appropriate devices. With respect to such questions we can consider utilizing our results to obtain an initial idea of the region of parameter space to explore more extensively by means of simulation or by improved analytical approximations, several of which are currently under development.

## 2. THE DIFFUSION APPROXIMATION.

We consider the cyclic queueing system (see Figure 1) which consists of two sequential stages. The system is assumed to serve a constant number,  $J$ , of programs ( $J \geq 2$ ), each of which goes through both stages in sequence and then returns to the first stage, this process being repeated continuously. It is assumed that after completion of CPU service a program moves instantaneously from stage 1 to the tail of the queue in stage 2, and after DTU service at that stage, back to the tail of the queue in stage 1. We shall suppose in addition that the queue in front of the CPU and the queue in front of the DTU are served according to a First-In-First-Out (FIFO) discipline. The assumption that  $J$  is a constant is an approximation that is justified by the common practice of operating such a system in a saturated mode.

In addition, we assume that the service times at the CPU and the DTU are mutually independent, and that CPU service times and DTU service times, respectively, are identically distributed.

We attempt here to give a brief intuitive account of our approximation. Let  $N_C(t)$  denote the number of programs present at the CPU at time  $t$ ; this includes those queued in addition to the program currently in service. Then if  $N_C(0) = 0$ ,

$$N_C(t) = A(t) - D(t) \tag{2.1}$$

where  $A(t)$  represents the number of arrivals at the CPU in  $(0, t)$ , and  $D(t)$  is the number CPU departures in  $(0, t)$ . If we neglect boundary effects at 0 and  $J$ ,  $A(t)$  and  $D(t)$  are independent

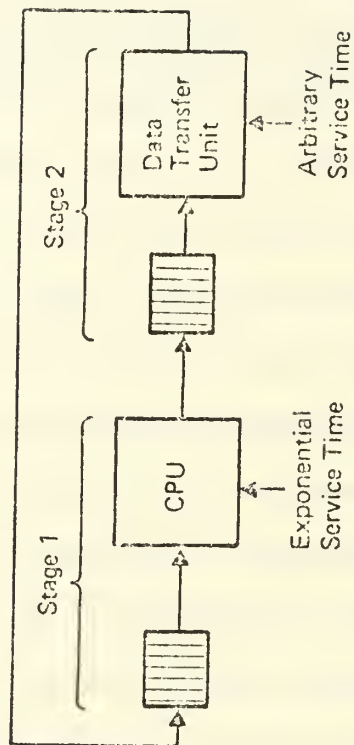


Figure 1. Cyclic Queue Model



renewal processes, so as  $t$  becomes large  $A(t)$  and  $D(t)$  are approximately normally distributed with means  $\frac{t}{E[D]}$  and  $\frac{t}{E[S]}$  and variances  $t \frac{\text{Var}[D]}{(E[D])^3}$  and  $t \frac{\text{Var}[S]}{(E[S])^3}$  respectively; see Cox [1], p. 40. It follows that  $N_C(t)$  is approximately normally distributed with mean  $\mu t = \left[ \frac{1}{E[D]} - \frac{1}{E[S]} \right] t$  and variance  $\sigma^2 t = \left[ \frac{\text{Var}[D]}{(E[D])^3} + \frac{\text{Var}[S]}{(E[S])^3} \right] t$ . We now approximate by replacing the difference of renewal processes by a diffusion (Wiener process) with drift  $\mu$  and infinitesimal variance  $\sigma^2$ . Thus,  $F(x,t)$ , the distribution of  $N_C(t)$ , satisfies the diffusion equation

$$\frac{\partial F}{\partial t} = -\mu \frac{\partial F}{\partial x} + \frac{\sigma^2}{2} \frac{\partial^2 F}{\partial x^2} \quad (2.2)$$

again approximately; cf. Newell [7], pp. 105-107. A reflecting boundary condition at  $x = 0$  must be imposed, for  $N_C(t) \geq 0$ , and another such boundary condition at  $x = J$  constrains  $N_C(t)$  to be  $\leq J$ . We require the solution to (2.3), subject to an initial distribution, e.g.

$$F(x,0) = \begin{cases} 1, & x \geq x_0 > 0 \\ 0, & x < x_0, \end{cases} \quad (2.3)$$

and boundary conditions

$$F(0+,t) \geq 0, \quad F(J,t) = 1. \quad (2.4)$$

For further details of the behavior of this approximation when  $J$  is infinite see Gaver [4]. For a closed form solution to (2.3) when  $J$  is infinite, see Newell [7].

### 3. THE STATIONARY DISTRIBUTION IN THE DIFFUSION APPROXIMATION.

The stationary or long-run distribution associated with our problem is  $F(x) = \lim_{t \rightarrow \infty} F(x, t)$  and satisfies

$$\frac{\sigma^2}{2} \frac{d^2 F}{dx^2} - \mu \frac{dF}{dx} = 0$$

for  $x > 0$ . Routine integrations lead to the solution

$$F(x) = A \left[ 1 - B e^{\frac{2\mu}{\sigma^2} x} \right]. \quad (3.1)$$

Invocation of the upper boundary condition provides that

$$1 = A \left[ 1 - B e^{\frac{2\mu}{\sigma^2} J} \right]. \quad (3.2)$$

According to (3.1) and (3.2) we have

$$F(x) = \frac{1 - B e^{\frac{2\mu}{\sigma^2} x}}{1 - B e^{\frac{2\mu}{\sigma^2} J}} \quad (3.3)$$

It remains to determine the constant  $B$ . We suggest several alternatives, and present numerical properties of these.

(1). Exact fit for  $J = 1$ .

If only one program occupies the system then renewal theory provides that (assuming  $E[D] = 1$ , as we shall throughout)

$$F(0+) = \frac{1}{1 + E[S]}. \quad (3.4)$$

If we set  $J = 1$ ,  $x = 0$  in (3.3) and equate the result to (3.4) the value of  $B$  is determined:

$$B = \frac{E[S]}{E[S] - e^{\frac{2\mu}{\sigma^2}}}$$

(2) Exact fit for  $J = 2$ .

If CPU service times are independently distributed copies of a random variable  $S$ , and likewise DTU service times are independent copies of the random variable  $D$ , then the long-run CPU utilization  $U$ , when  $J = 2$ , given by

$$U = 1 - F(0+) = \frac{E[S]}{E[\text{Max}(S,D)]} \quad (3.5)$$

This was pointed out by J. Gecsei (private communication). This can be seen as follows. Consider the sequence of times  $\{\tau_K\}$  at which either i) the CPU is idle, a DTU service has just been completed and the served program has moved to the CPU stage queue or ii) the DTU is idle, a CPU service has just been completed and the served program has moved to the DTU stage queue. These  $\{\tau_K\}$  are regeneration points in the process, the mean regeneration time being  $E[\text{Max}(S,D)]$ . Since the mean amount of time that the CPU is busy between regeneration points is  $E(S)$ , (3.5) follows from a basic limit theorem for regenerative processes; see Feller [2], p. 365. Making use of (3.3) we obtain

$$B = \frac{E[S]}{E[\text{max}(S,D)] \left[ 1 - e^{\frac{4\mu}{\sigma^2}} \right] + E[S] e^{\frac{4\mu}{\sigma^2}}} . \quad (3.6)$$

(3) Exact fit for large  $J$  ( $J \rightarrow \infty$ ).

Of most practical interest is the case of large  $J$ . The diffusion approximation may also be expected to work best when boundaries are visited infrequently, and this implies large  $J$  and near equality of  $E[S]$  and  $E[D]$ .

In practice  $\mu < 0$ , so we may allow  $J \rightarrow \infty$  in (3.3) to discover that

$$F(x) = \lim_{J \rightarrow \infty} \frac{1 - B e^{\frac{2\mu}{\sigma^2} x}}{1 - B e^{\frac{2\mu}{\sigma^2} J}} = 1 - B e^{\frac{2\mu}{\sigma^2} x} \quad (3.7)$$

and hence for  $J = \infty$ ,

$$F(0+) = 1 - B \quad (3.8)$$

But for arbitrary queueing systems of the type under consideration (see Takács [10], p. 142),

$$\lim_{t \rightarrow \infty} P\{N_C(t) = 0\} = 1 - \rho, \quad \text{if } \rho < 1$$

where  $\rho$  is the traffic intensity parameter:

$$\rho = \frac{E[S]}{E[D]} < 1$$

when  $J = \infty$ . Consequently we put  $B = \rho$  and find

$$F(0+) = \frac{1 - \rho}{1 - \rho e^{\frac{2\mu}{\sigma^2} J}} \quad (3.9)$$

This simple expression promises to give a good approximation when  $J$  is reasonably large and  $\mu$  is not far from zero but has a negative sign. The quality of the approximations may be judged by making reference to the numerical examples that follow.

#### 4. NUMERICAL EXAMPLES.

In [8], Shedler has tabulated CPU utilization, namely the long-run probability that the CPU is busy; the latter depends upon the DTU distribution and the degree of multiprogramming,  $J$ . The numbers obtained are the result of a semi-Markov analysis; the success of the latter seems to depend rather crucially upon the assumption of an exponential service time at the CPU.

Our diffusion approximation is capable of supplying figures for CPU utilization--an explicit formula follows directly from (3.9) by subtraction from unity:

$$\text{CPU utilization} = \rho \left\{ \frac{1 - e^{-\frac{2\mu}{\sigma^2} J}}{1 - \rho e^{-\frac{2\mu}{\sigma^2} J}} \right\} \quad (4.1)$$

when  $\rho < 1$ . Here we compare the "exact" numbers obtained in [8] with those delivered by (4.1). Since the usual system is likely to have a relatively high degree of multiprogramming, we shall first discuss Method 3 for fitting  $B$  (exact fit for  $J = \infty$ ). Refer to Table 1. Evidently the Diffusion (Diff) and semi-Markov (S-M) figures agree quite closely. The numerical comparisons suggest that the diffusion approach provides an underestimate in the case of an exponential DTU. That this will always be the case may be shown analytically. Before doing so, we shall compare some of the results of Table 1 to the numbers obtained when Method 2 is used (exact fit for  $J = 2$ ). This comparison is illustrated in Table 2. Apparently the procedure of fitting  $J = 2$  exactly does not agree as well with the S-M calculations as does the  $J = \infty$  exact fit. When we recollect that explicit

calculation of  $E[\max(S,D)]$  for general distribution of  $S$  and  $D$  is a bit troublesome, it becomes easier to recommend the method of fit obtained by Method 3.

# 5. THE DIFFUSION APPROXIMATION TO THE MARKOVIAN CYCLIC SYSTEM.

In this section we shall show that the tendency for the diffusion (in the fit of Method 3) to underestimate CPU utilization in the exponential CPU and DTU (entirely Markov) case is no accident.

Denote the mean of  $S$ , the exponential service time at the CPU, by  $E[S] = \lambda^{-1}$ ; again  $E[D] = 1$ . Then explicit probability balance equations may be written:

$$\lambda p_0 = p_1$$

$$(\lambda+1)p_i = \lambda p_{i-1} + p_{i+1} \quad 1 \leq i \leq J-1$$

$$p_J = \lambda p_{J-1},$$

where  $p_j$  is the long-run probability of  $j$  programs at the DTU.

The solution is, expressed in terms of  $\rho = \lambda^{-1}$ ,

$$p_i = \frac{1 - \rho}{1 - \rho^{J+1}} \rho^{J-i}.$$

Thus

$$\text{CPU utilization} = U_e = 1 - p_J$$

where the subscript signifies "exact;" the corresponding figure for the diffusion is  $U_d$ , and the figures for idleness are  $I_e = p_J$ , and  $I_d = 1 - U_d$ , respectively. It will be shown that

$$I_d > I_e$$

or



$$\frac{1 - \rho}{1 - \rho e^{\frac{2\mu}{\sigma^2} J}} > \frac{1 - \rho}{1 - \rho^{J+1}}$$

Now in the present case  $\frac{2\mu}{\sigma^2} = 2\left(\frac{\rho-1}{\rho+1}\right)$ , and an easily verified inequality ( $\ln x < 2\left(\frac{x-1}{x+1}\right)$  for  $0 < x < 1$ ) shows that  $\rho < \exp\left[2\left(\frac{\rho-1}{\rho+1}\right)J\right]$  which verifies the assertion. It may also be shown that the error committed by use of the diffusion approximation decreases as  $J$  increases, but not to zero. The approximation improves, however, as  $E[S] \rightarrow E[D]$ .

## 6. DIFFUSION APPROXIMATION FOR AN EXTENSION OF THE MODEL.

In this section we shall indicate how a diffusion approximation for CPU utilization can be obtained when the assumption of identically distributed CPU service times is relaxed to permit the  $J$  programs to have different page-fault characteristics. Specifically we shall now assume that the sequence of CPU service times are mutually independent, and that the CPU service times of program  $i$  are identically distributed as a random variable  $S_i (1 \leq i \leq J)$ . The diffusion approximation proposed will again be of the form (4.1), and we seek appropriate expressions for  $\mu$ ,  $\sigma^2$  and  $\rho$ .

It is easily verified that the successive CPU service times  $S$  in the cyclic queueing system are a semi-Markov process, and that within this semi-Markov process  $Z(t)$ , the total number of page-faults in time  $t$ , is a cumulative stochastic process; see Cox [1]. It then follows from cumulative process theory that

$$\mu_z \equiv \lim_{t \rightarrow \infty} \frac{E\{Z(t)\}}{t} = \frac{J}{E[S_1] + \dots + E[S_J]} \quad (6.1)$$

$$\begin{aligned} \sigma_z^2 &\equiv \lim_{t \rightarrow \infty} \frac{\text{Var}\{Z(t)\}}{t} = \frac{J^2(\text{Var}[S_1] + \dots + \text{Var}[S_J])}{(E[S_1] + \dots + E[S_J])^3} \\ &= \frac{(\text{Var}[S_1] + \dots + \text{Var}[S_J])}{\left( \frac{E[S_1] + \dots + E[S_J]}{J} \right)^3} \end{aligned} \quad (6.2)$$

To establish the drift and infinitesimal variance for the diffusion approximation put

$$\mu = 1 - \mu_z = 1 - \frac{J}{E[S_1] + \dots + E[S_J]} , \quad (6.3)$$

$$\sigma^2 = \frac{\text{Var}[D]}{E[D]^3} + \sigma_z^2 = \frac{\text{Var}[D]}{(E[D])^3} + \frac{\left( \frac{\text{Var}[S_1] + \dots + \text{Var}[S_J]}{J} \right)}{\left( \frac{E[S_1] + \dots + E[S_J]}{J} \right)^3} \quad (6.4)$$

and

$$\rho = \frac{1}{\mu_z} = \frac{E[S_1] + \dots + E[S_J]}{J} \quad (6.5)$$

One can also assume that each program's DTU service time comes from a specific distribution, and generalize the above expressions accordingly in an obvious way.

## 7. CONCLUSIONS AND DIRECTIONS FOR FURTHER WORK.

Although certain of the cyclic queue problems relevant to multiprogramming studies may be solved by conventional methods (semi-Markov processes, or the "phase" approach described in Morse [6]), the results are nearly always cumbersome and difficult to compute. The approximate approach exhibited here furnishes simple formulas that may be evaluated by hand computation. The accuracy of the approximation seems quite adequate for the cases studied. However, efforts to improve the approximation are suggested for the following reasons.

- (1) Data analysis of actual CPU service times indicate that the latter are apt to be somewhat more skewed than the exponential.
- (2) The cyclic queue model discussed here may also be used to represent a reliability situation. In this interpretation components of subsystems take the place of programs. A DTU service time represents the service life of a component, and CPU service is identified with a repair; in this model there is only one repair facility. Finally,  $J - 1$  is the number of spare components in the system. One question of interest concerns the long-run probability that the DTU is "busy," equivalent to the probability that all  $J$  items in the system are not undergoing repair. This "availability" figure is approximated by  $F(J-1)$ . Another question relates to the probability that if initially (i) a new component is in operation (at the DTU), and (ii) the  $J - 1$  spares are queued behind the latter, then the time until the DTU becomes idle for the first time exceeds  $t$ . The latter probability may

be approximated by solving a diffusion equation, this time with one reflecting and one absorbing barrier. The details remain to be worked out, and the quality of the approximation evaluated.

## DTU Service Distribution (Mean = 1)

E[S]	Erlang - 1 (exponential)		Erlang - 2		Erlang - 3		Erlang - 4		Erlang - 5		Erlang - $\infty$ (constant)	
	S-M	Diff.	S-M	Diff.	S-M	Diff.	S-M	Diff.	S-M	Diff.	S-M	Diff.
2	.238	.233	.243	.237	.245	.238	.246	.239	.247	.239	.249	.241
3	.247	.245	.249	.247	.250	.247	.250	.247	.250	.247	.250	.248
4	.25	.249	.249	.250	.249	.250	.249	.250	.249	.250	.249	.250
5	.250	.250	.250	.250	.250	.250	.250	.250	.250	.250	.250	.250
6	.250	.250	.250	.250	.250	.250	.250	.250	.250	.250	.250	.250
2	.429	.424	.444	.444	.451	.451	.455	.454	.458	.456	.468	.464
3	.467	.464	.480	.476	.485	.480	.487	.482	.489	.483	.494	.487
4	.50	.484	.482	.493	.490	.495	.492	.496	.493	.497	.493	.499
5	.492	.491	.497	.495	.498	.497	.499	.497	.499	.497	.500	.498
6	.496	.495	.499	.498	.500	.499	.500	.499	.500	.499	.500	.499
2	.568	.566	.591	.608	.601	.623	.606	.631	.610	.636	.626	.655
3	.634	.633	.660	.666	.670	.677	.676	.683	.679	.686	.694	.700
4	.75	.672	.672	.695	.697	.704	.706	.709	.710	.711	.712	.723
5	.696	.695	.716	.716	.722	.722	.726	.725	.728	.727	.736	.734
6	.712	.711	.728	.727	.733	.732	.736	.734	.737	.736	.743	.741
2	.590	.589	.614	.635	.624	.652	.630	.661	.633	.666	.651	.688
3	.661	.661	.689	.697	.700	.710	.701	.717	.709	.721	.726	.737
4	.80	.702	.702	.728	.732	.738	.742	.743	.747	.746	.750	.762
5	.729	.729	.751	.753	.759	.761	.763	.764	.766	.767	.776	.776
6	.747	.747	.766	.766	.773	.773	.776	.776	.778	.778	.786	.784
2	.631	.631	.656	.685	.667	.705	.673	.715	.677	.721	.694	.748
3	.709	.709	.739	.753	.751	.769	.757	.777	.762	.782	.780	.803
4	.90	.756	.756	.785	.792	.796	.806	.802	.812	.806	.816	.832
5	.787	.787	.813	.818	.823	.829	.828	.834	.831	.837	.845	.851
6	.808	.808	.832	.835	.841	.844	.846	.849	.848	.852	.860	.863

Table 1. CPU Utilization Comparison  
Based on fitting B for  $J=\infty$ .

(CPU exponential) mean  $E[S]$

## DTU Service Distribution (Mean = 1)

J	E[S]	S-M	Exponential Diff. (Fit J=2)	Diff. (Fit J= $\infty$ )	S-M	Constant Diff. (Fit J=2)	Diff. (Fit J= $\infty$ )
2		.238	.238	.233	.249	.249	.241
3		.247	.251	.245	.250	.256	.248
4	.25	.249	.254	.249	.250	.258	.250
5		.250	.255	.250	.250	.258	.250
6		.250	.256	.250	.250	.259	.250
2		.429	.429	.424	.468	.468	.464
3		.467	.468	.464	.494	.492	.487
4	.50	.484	.487	.482	.499	.500	.495
5		.492	.495	.491	.500	.503	.498
6		.496	.500	.495	.500	.504	.499
2		.568	.568	.566	.626	.626	.655
3		.634	.634	.633	.694	.673	.700
4	.75	.672	.673	.672	.723	.696	.722
5		.696	.696	.695	.736	.708	.734
6		.712	.712	.711	.743	.716	.741
2		.590	.590	.589	.651	.651	.688
3		.661	.661	.661	.726	.702	.737
4	.80	.702	.703	.702	.759	.730	.762
5		.729	.729	.729	.776	.745	.776
6		.747	.747	.747	.786	.755	.784
2		.631	.631	.631	.694	.694	.748
3		.709	.709	.709	.780	.757	.803
4	.90	.756	.756	.756	.821	.791	.832
5		.787	.787	.787	.845	.813	.851
6		.808	.808	.808	.860	.828	.863

Table 2. CPU Utilization Comparison

(CPU exponential mean E[S])

## REFERENCES

- [1] Cox, D. R., *Renewal Theory*, Methuen Monograph, London: Methuen and Co., Ltd., 1962.
- [2] Feller, W., *An Introduction to Probability Theory and its Applications*, Vol. II, John Wiley and Sons, New York, 1966.
- [3] Gaver, D. P., "Probability models for multiprogramming computer systems," J. ACM, 14 (3), 423-238 (1967).
- [4] Gaver, D. P., "Diffusion approximations and models for certain congestion problems," J. Appl. Prob. 5, 607-623 (1968).
- [5] Lewis, P. A. W. and Shedler, G. S., "A cyclic-queue model of system overhead in multiprogrammed computer systems," J. ACM, 18 (2), 199-220 (1971).
- [6] Morse, P. M., *Queues, Inventories and Maintenance*, John Wiley and Sons, New York, 1958.
- [7] Newell, G. F., *Applications of Queueing Theory*, Chapman and Hall, Ltd., London, 1971.
- [8] Shedler, G. S., "A cyclic-queue model of a paging machine," IBM Research Report RC-2814, IBM Watson Research Center, Yorktown Heights, New York (March 1970).
- [9] Shedler, G. S., "A queueing model of a multiprogrammed computer with a two-level storage system," IBM Research Report RJ-836, IBM Research Laboratory, San Jose, Calif. (March 1971), to appear Comm. ACM.
- [10] Takács, L., *Introduction to the Theory of Queues*, Oxford University Press, New York, 1962.



## INITIAL DISTRIBUTION LIST

	No. Copies
Defense Documentation Center (DDC) Cameron Station Alexandria, Virginia 22314	20
Library Naval Postgraduate School Monterey, California 93940	2
Dean of Research Administration Naval Postgraduate School Monterey, California 93940	1
Library Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	3
Professor D. R. Cox Department of Mathematics Imperial College London, England	1
Professor B. Epstein Israel Institute of Technology Technion City Haifa, Israel	1
Professor P. A. P. Moran Department of Statistics Australian National University P.O. Box 4 Canberra, Australia 2600	1
Professor Daryl Daley Department of Statistics Australian National University P.O. Box 4 Canberra, Australia 2600	1
Marvin Denicoff Office of Naval Research Arlington, Virginia 22217	1
Dr. Thomas Varley Office of Naval Research Arlington, Virginia 22217	1

Professor U. N. Bhat Computer Science Center Institute of Technology Southern Methodist University Dallas, Texas 75222	1
Dr. Richard Nance Computer Science Center Institute of Technology Southern Methodist University Dallas, Texas 75222	1
Professor John Lehoczky Statistics Department Carnegie-Mellon University Schenley Park Pittsburgh, Pa. 15213	1
Professor Donald McNeill Statistics Department Princeton University Princeton, New Jersey 08540	1
Professor Carl Harris School of Engineering George Washington University Washington, D.C. 20006	1
Dr. Daniel Heyman Bell Telephone Inc. Crawford Corner Road Holmdel, New Jersey 07733	1
Dr. Ronald Wolff Operations Research Center Massachusetts Institute of Technology Cambridge, Massachusetts 02139	1
Dr. Bruce McDonald Office of Naval Research Arlington, Virginia 22217	1
Professor Walter Smith Statistics Department University of North Carolina Chapel Hill, North Carolina 27514	1

Dr. Glenn Bacon Systems Department IBM Research Monterey and Cottle Roads San Jose, California 95111	1
Professor Leonard Kleinrock Department of Computer Science University of California Los Angeles, California 90024	1
Dr. Bill Mitchell Department of Management Sciences California State College Hayward, California 94542	1
Dr. Mathew Sobel Administrative Science Yale University New Haven, Connecticut 06520	1
Dr. P.A.W. Lewis	1
Dr. Kneale Marshall	1
Dr. R. Butterworth	1
Dr. Paul Milch	1
Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	
Dr. Donald P. Gaver Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	10



Security Classification

## DOCUMENT CONTROL DATA - R &amp; D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

ORIGINATING ACTIVITY (Corporate author)

Naval Postgraduate School  
Monterey, California

2a. REPORT SECURITY CLASSIFICATION

UNCLASSIFIED

2b. GROUP

REPORT TITLE

Processor Utilization in Multiprogramming Systems  
Via Diffusion Approximations

3. DESCRIPTIVE NOTES (Type of report and, inclusive dates)

Technical Report

4. AUTHOR(S) (First name, middle initial, last name)

Donald P. Gaver and Gerald S. Shedler

5. REPORT DATE

May 1972

7a. TOTAL NO. OF PAGES

27

7b. NO. OF REFS

10

6. CONTRACT OR GRANT NO.

9a. ORIGINATOR'S REPORT NUMBER(S)

8. PROJECT NO.

c.

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned  
this report)

d.

9. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

This report describes an approximation for the central processor (CPU) utilization in a multiprogramming computer system. The approximation is based on use of the mathematical theory of diffusion; its adequacy is checked numerically and found to be good.

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Computer systems						
Multiprogramming						
Probability						
Queues						
Waiting Lines						
Utilization						

U146010

DUDLEY KNOX LIBRARY - RESEARCH REPORTS



5 6853 01058126 7

